What is claimed is:

1. A method for performing interpolation of a point of interest producing an approximated function value of said point of interest using an input data set comprising:

determining distances between said point of interest and points in said input data set;

sorting said distances;

selecting a predetermined number of points in said input data set which are closest to said point of interest in accordance with said distances;

determining whether said point of interest is one of: enclosed within a simplex and on a boundary of the simplex, said simplex formed by a combination of n+1 points selected from said predetermined number of points, said n+1 points forming vertices of said simplex, n being a dimension of points in said input data set; and

if there is a simplex enclosing said point of interest or including said point of interest on its boundary, fitting a linear surface to said vertices of said simplex in accordance with a linear function, and evaluating said linear function at said point of interest to determine said approximated function value representing the approximation of said linear function at said point of interest.

2. The method of Claim 1, further comprising:

examining combinations of n+1 points selected from said predetermined number of points to form candidate simplices.

3. The method of Claim 2, further comprising:

determining that there is at least one simplex of said candidate simplices that either enclose said point of interest or include said point of interest on a boundary thereof.

4. The method of Claim 2, further comprising, for each of said candidate simplices:

shifting a coordinate system of said each candidate simplex to have a first vertex v(1) of said each candidate simplex as a point of origin;

forming an n x n matrix A represented as:

$$A = \left[\bar{v}(2), \bar{v}(3), \ldots, \bar{v}(n+1)\right]$$

wherein each $\bar{v}(j)$ represents a shifted vertex corresponding to an original vertex shifted in accordance with said point of origin v(1), for each vertex v(j), j=2, .., n+1, said shifted vertex for each vertex j, j=2 .. n+1 denoted as:

$$\bar{v}(j) = v(j) - v(1) \quad \text{and}$$

defining for the point of interest a corresponding shifted point of interest:

$$\bar{x} = \tilde{x} - v(1)$$

where $\tilde{x}$ represents said point of interest; and

solving a linear equation Aa= $\bar{x}$ for unknown vector "a".

5. The method of Claim 4, further comprising:

determining whether said point of interest is one of: enclosed within said each candidate simplex and included on a boundary of said each candidate simplex, by determining if the vector "a" satisfies:

$$a_i \geq 0 \text{ and } \sum_{i=1}^{n} a_i \leq 1$$

in which $a_1, .. a_n$ are entries in the vector "a".

6. The method of Claim 5, further comprising:

fitting a linear surface to vertices of at least one of said each candidate simplices by solving a linear equation:

$$b^T B = u^T \text{ for "b",}$$

where values of said linear function at points v(j) are denoted as u(j), j = 1 .. n+1 such that "u" is a vector defined in terms of all the u(j)'s as a transpose matrix:

$$u = [u(1), u(2), \ldots, u(n+1)]^T$$

and B is an n+1 by n+1 matrix represented as:

$$B = \begin{bmatrix} v(1) & v(2) & \ldots v(n+1) \\ 1 & 1 & 1 \end{bmatrix}$$

7. The method of Claim 1, wherein, said determining whether said point of interest is one of: enclosed within a simplex and included on a boundary of the simplex comprises:

expressing said point of interest as a linear combination of n vertices of said simplex having translated coordinates with vertex "n+1" of the simplex as a point of origin; and

determining if said linear combination is convex.

8. The method of Claim 7, wherein said determining if said linear combination is convex further comprises:

determining if all coefficients of said linear combination are positive and sum to less than 1.

9. The method of Claim 2, further comprising:

determining that none of said candidate simplices includes said point of interest such that said point of interest is not enclosed by a candidate simplex and is not on a boundary thereof.

10. The method of Claim 9, further comprising:

performing an alternate action.

11. The method of Claim 10, wherein said alternate action is at least one of: an alternate interpolation technique, increasing said predetermined number of points, and performing linear extrapolation.

12. The method of Claim 10, wherein said alternate action is performing linear extrapolation, and said method further comprising:

forming a simplex having n+1 vertices that are the n+1 nearest neighbors of said point of interest.

13. The method of Claim 10, wherein said alternate action is performing linear extrapolation, and said method further comprising:

selecting a simplex from said candidate simplices having a smallest violation of all said candidate simplices.

14. The method of Claim 1, wherein said input data set is non-uniform.

15. The method of Claim 1, wherein said input data set if not colinear.

16. The method of Claim 6, further comprising:

determining if said input data set is colinear; and

if said input data set is colinear, adding a predetermined value to said input data such that a resulting input data set is not colinear.

17. The method of Claim 1, further comprising:

adding said point of interest and said approximated function value to said input data set producing a revised data set; and

using said revised data set with a neural network.

18. The method of Claim 17, further comprising:

using said revised data to train said neural network.

19. The method of Claim 17, further comprising:

using said revised data to verify said neural network.

20. The method of Claim 1, further comprising:

estimating helicopter gross weight.

21. A computer program product for performing interpolation of a point of interest producing an approximated function value of said point of interest using an input data set comprising:

executable code that determines distances between said point of interest and points in said input data set;

executable code that sorts said distances;

executable code that selects a predetermined number of points in said input data set which are closest to said point of interest in accordance with said distances;

executable code that determines whether said point of interest is one of: enclosed within a simplex and on a boundary of the simplex, said simplex formed by a combination of n+1 points selected from said predetermined number of points, said n+1 points forming vertices of said simplex, n being a dimension of points in said input data set; and

executable code that, if there is a simplex enclosing said point of interest or including said point of interest on its boundary, fits a linear surface to said vertices of said simplex in accordance with a linear function, and evaluating said linear function at said point of interest to determine said approximated function value representing the approximation of said linear function at said point of interest.

22. The computer program product of Claim 21, further comprising:

executable code that examines combinations of n+1 points selected from said predetermined number of points to form candidate simplices.

23. The computer program product of Claim 22, further comprising:

executable code that determines that there is at least one simplex of said candidate simplices that either enclose said point of interest or include said point of interest on a boundary thereof.

24. The computer program product of Claim 22, further comprising, executable code that, for each of said candidate simplices:

shifts a coordinate system of said each candidate simplex to have a first vertex v(1) of said each candidate simplex as a point of origin;

forms an n x n matrix A represented as:

$$A = \left[\bar{v}(2), \bar{v}(3), \ldots, \bar{v}(n+1)\right]$$

wherein each $\bar{v}(j)$ represents a shifted vertex corresponding to an original vertex shifted in accordance with said point of origin v(1), for each vertex v(j), j=2, .., n+1, said shifted vertex for each vertex j, j=2 .. n+1 denoted as:

$$\bar{v}(j) = v(j) - v(1) \qquad \text{and}$$

defines for the point of interest a corresponding shifted point of interest:

$$\bar{x} = \tilde{x} - v(1)$$

where $\tilde{x}$ represents said point of interest; and

solves a linear equation Aa= $\bar{x}$ for unknown vector "a".

25. The computer program product of Claim 24, further comprising:

executable code that determines whether said point of interest is one of: enclosed within said each candidate simplex and included on a boundary of said each candidate simplex, by determining if the vector "a" satisfies:

$$a_i \geq 0 \text{ and } \sum_{i=1}^{n} a_i \leq 1$$

in which $a_1, .. a_n$ are entries in the vector "a".

26. The computer program product of Claim 25, further comprising:

executable code that fits a linear surface to vertices of at least one of said each candidate simplices by solving a linear equation:

$$b^T B = u^T \text{ for "b",}$$

where values of said linear function at points v(j) are denoted as u(j), j = 1 .. n+1 such that "u" is a vector defined in terms of all the u(j)'s as a transpose matrix:

$$u = [u(1), u(2), \ldots, u(n+1)]^T$$

and B is an n+1 by n+1 matrix represented as:

$$B = \begin{bmatrix} v(1) & v(2) & \ldots v(n+1) \\ 1 & 1 & 1 \end{bmatrix}$$

27. The computer program product of Claim 21, wherein, said executable code that determines whether said point of interest is one of: enclosed within a simplex and included on a boundary of the simplex comprises:

executable code that expresses said point of interest as a linear combination of n vertices of said simplex having translated coordinates with vertex "n+1" of the simplex as a point of origin; and

executable code that determines if said linear combination is convex.

28. The computer program product of Claim 27, wherein said executable code that determines if said linear combination is convex further comprises:

executable code that determines if all coefficients of said linear combination are positive and sum to less than 1.

29. The computer program product of Claim 22, further comprising:

executable code that determines that none of said candidate simplices includes said point of interest such that said point of interest is not enclosed by a candidate simplex and is not on a boundary thereof.

30. The computer program product of Claim 29, further comprising:

executable code that performs an alternate action.

31. The computer program product of Claim 30, wherein said alternate action is at least one of: an alternate interpolation technique, increasing said predetermined number of points, and performing linear extrapolation.

32. The computer program product of Claim 30, wherein said alternate action is performing linear extrapolation, and said computer program product further comprising:

executable code that forms a simplex having n+1 vertices that are the n+1 nearest neighbors of said point of interest.

33. The computer program product of Claim 30, wherein said alternate action is performing linear extrapolation, and said computer program product further comprising:

executable code that selects a simplex from said candidate simplices having a smallest violation of all said candidate simplices.

34. The computer program product of Claim 21, wherein said input data set is non-uniform.

35. The computer program product of Claim 21, wherein said input data set if not colinear.

36. The computer program product of Claim 26, further comprising:

executable code that determines if said input data set is colinear; and

executable code that, if said input data set is colinear, adds a predetermined value to said input data such that a resulting input data set is not colinear.

37. The computer program product of Claim 21, further comprising:

executable code that adds said point of interest and said approximated function value to said input data set producing a revised data set; and

executable code that uses said revised data set with a neural network.

38. The computer program product of Claim 37, further comprising:

executable code that uses said revised data to train said neural network.

39. The computer program product of Claim 18, further comprising:

executable code that uses said revised data to verify said neural network.

40. The computer program product of Claim 21, further comprising executable code to estimate helicopter gross weight.